



Function and Usage of the ModBus RTU Slave Library ModBusSlave.lib

Document Version 1.0

CONTENT

1	PRELIMINARY REMARKS	3
2	SHORT DESCRIPTION	4
3	USAGE IN A PROJECT	5
3.1	Create new CoDeSys project	5
3.2	Including the required libraries	5
3.3	Using the function block instances in PLC_PRG	5
3.4	Parameterizing of the ModSlave1 instances	6
4	COMPILATION AND PROCESSING OF THE PROGRAM	7
5	SUPPORTED MODBUS FUNCTIONS	8
	CHANGE HISTORY	9

1 Preliminary Remarks

Based on the system library SysLibCom.lib, which is provided by the CoDeSys runtime system for the use of a serial interface, a protocol library for the data exchange via Modbus RTU was developed.

This library allows, with a very simple configuration and parameterizing, the data exchange between a Modbus-Master and one or several Modbus-Slaves.

This documentation describes the modules provided by the library as well as their function and usage.

2 Short Description

Subsequently the realization of the library will be described in key points:

The library consists of two user function blocks which, depending on the currently used target system, must be used alternatively. Also the library contains three data structures which are only used within the user function blocks.

The user will utilize one of the function blocks, depending on the currently used target system. This differentiation is necessary because depending on the target the received WORD data have either to be swapped or not to be swapped.

For systems with Motorola byte order the user module to be used is „ModbusSlaveMSB“. For systems with Intel byte order the user module to be used is „ModbusSlaveLSB“. The parameters and the use of the modules however is the same for both system types.

Function blocks which have to be programmed by the user:

Nr	Sort	Type	Name	Task
1	User	FB	ModbusSlaveLSB alternatively: ModbusSlaveMSB	Base module for the parameterization of a serial interface and for the execution of the Modbus protocol. The data provided by the controller are handed over to the module as a pointer on the project data.

For the access on the serial interface the protocol library uses the CoDeSys system library „SysLibCom.Lib“. This library is the runtime-independent abstraction for the use of the serial interface. Additionally library „Standad.Lib“ is attracted. Both libraries are included automatically with the protocol library.

3 Usage in a project

The usage of the protocol library is very easy. The required procedure in a CoDeSys project will be described in detail in the following, based on the assumption that a new project is to be created.

3.1 Create new CoDeSys project

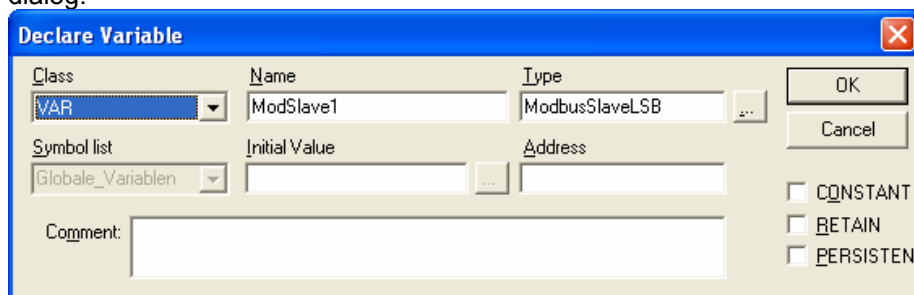
- Start CoDeSys.
- Create new project via **File/New**.
- Select the appropriate target setting (Target Settings), e.g. CoDeSys SP for Windows NT Realtime.
- Create a new POU "PLC_PRG" (programming language, FBD in the example).

3.2 Including the required libraries

- Open Library Manager via **Window/Library Manager**.
- Include library ModbusSlave.LIB via menu **Insert/Additional library...**
- The libraries STANDARD.LIB and SYSLIBCOM.LIB are included automatically. If not, repeat the preceding step for those libraries.

3.3 Using the function block instances in PLC_PRG

- For a cyclic processing of the Modbus-Slave function an instance of the Modbus-Slave function block must be called in the main POU. For this purpose a module is inserted and parameterized in PLC_PRG:
- Place the cursor in the first network of PLC_PRG behind the "???" and insert a module via menu Insert/Box.
- Replace keyword "AND" in the box by "ModbusSlaveLSB" resp. "ModbusSlaveMSB". Insert the instance name by replacing "???" e.g. by "ModSlave1".
- When leaving the edit field for the instance name, CoDeSys automatically opens the declaration dialog:

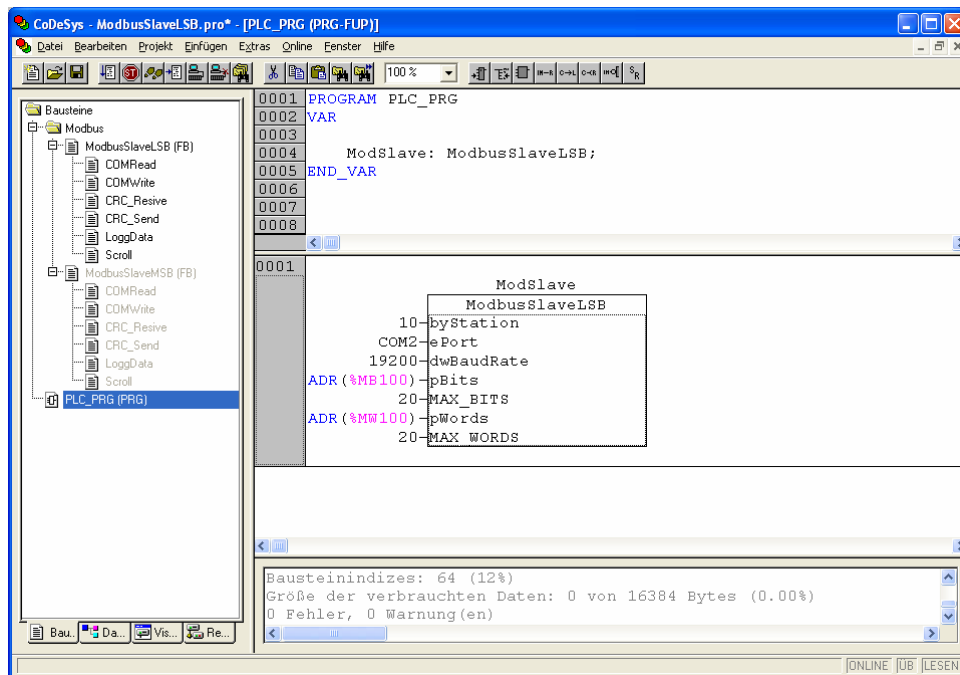


- In our example we assume that the instance should be declared as a local variable within PLC_PRG. So we can now simply confirm the settings and close the dialog by OK.

3.4 Parameterizing of the ModSlave1 instances

For a correct processing the module must get assigned some parameters..

Input	Meaning	Example
byStation	Parameters for the station address of the slave. Permissible values: 1..128	1
ePort	Parameter for the interface. Permissible values, depending on the target system: COM1...COM8	COM2
dwBaudRate	Used baud rate	19200
pBits	Address of the data range for the MODBUS bits. In the example the flags %MB100 to %MB119 are used.	ADR(%MB100)
MAX_BITS	Number of BYTES, available as bits for the MODBUS Slave. In the example 20 Byte = 160 Bits are parameterized.	20
pWords	Address of the data range for the MODBUS Words. In the example the flags %MW100 to %MW119 are used.	ADR(%MW100)
MAX_WORDS	Number of WORDS, available for the MODBUS Slave.	20



4 Compilation and processing of the program

After the parameterization of the function block the program can be compiled via **Project/Build** and afterwards can be load to the controller via **Online/Login** and can be started there via **Online/Start**.

The modules allow an automatic data exchange between the Modbus-Master and the PLC. Hereby the data of the parameterized data areas either are read or written by the connected Modbus-Master. Instead of the flags used in the example of course also own data structures can be created, the addresses of which then will be handed over to the ModbusSlaveMSB/LSB module.

5 Supported Modbus Functions

The Modbus-Slave library supports the following functions:

Code	Description
1 / 2	Read Bits - Simultaneous reading of several bits as WORD
3 / 4	Read Register – Simultaneous reading of one or several WORD's with subsequent addresses
5	Write Bits - Writes a single bit
6	Write Register - Writes a single WORD
16	Write Register - Simultaneous writing of one or several WORD's with subsequent addresses

Change History

Version	Description	Editor	Date
0.1	Issued	CS	01.02.2005
1.0	Review german version	AF	09.09.2005
1.0	Translation according to reviewed german version; Release	MN	12.09.2005